



A Reflective Framework for Performance Management (REFORM) of Real-Time Hybrid Simulation

Amin Maghreh^{1*}, **Yuguang Fu**¹, **Herta Montoya**², **Johnny Condori**², **Zixin Wang**²,
Shirley J. Dyke^{1,2} and **Arturo Montoya**³

¹ School of Mechanical Engineering, Purdue University, West Lafayette, IN, United States, ² Lyles School of Civil Engineering, Purdue University, West Lafayette, IN, United States, ³ Department of Civil and Environmental Engineering, The University of Texas at San Antonio, San Antonio, TX, United States

OPEN ACCESS

Edited by:

Chia-Ming Chang,
National Taiwan University, Taiwan

Reviewed by:

Takehiko Asai,
University of Tsukuba, Japan
Richard Christenson,
University of Connecticut,
United States

***Correspondence:**

Amin Maghreh
amaghreh@purdue.edu

Specialty section:

This article was submitted to
Computational Methods in Structural
Engineering,
a section of the journal
Frontiers in Built Environment

Received: 02 June 2020

Accepted: 18 August 2020

Published: 25 September 2020

Citation:

Maghreh A, Fu Y, Montoya H, Condori J, Wang Z, Dyke SJ and Montoya A (2020) A Reflective Framework for Performance Management (REFORM) of Real-Time Hybrid Simulation. *Front. Built Environ.* 6:568742. doi: 10.3389/fbui.2020.568742

Currently, the lack of (1) a sufficiently integrated, adaptive, and reflective framework to ensure the safety, integrity, and coordinated evolution of a real-time hybrid simulation (RTHS) as it runs, and (2) the ability to articulate and gauge suitable measures of the performance and integrity of an experiment, both as it runs and *post-hoc*, have prevented researchers from tackling a wide range of complex research problems of vital national interest. To address these limitations of the current state-of-the-art, we propose a framework named Reflective Framework for Performance Management (REFORM) of real-time hybrid simulation. REFORM will support the execution of more complex RTHS experiments than can be conducted today, and will allow them to be configured rapidly, performed safely, and analyzed thoroughly. This study provides a description of the building blocks associated with the first phase of this development (REFORM-I). REFORM-I is verified and demonstrated through application to an expanded version of the benchmark control problem for real-time hybrid simulation.

Keywords: real-time hybrid simulation, run-time sensitivity indicator, self-tuning robust control, run-time stability threshold, RTHS, REFORM, real-time control, RTHS framework

INTRODUCTION

Engineers in the coming decades will need to push the boundaries in infrastructure design. New materials, additive construction methods, smart materials and dampers, bio-inspired designs, etc. are being developed to support this vision for an incredible infrastructure of tomorrow. However, realizing the implementation of these novel materials and structures first requires that we provide evidence that they can perform at levels that go well beyond present-day expectations. They must be able to withstand extreme loads under uncertain conditions; they must be proven to sustain the severe loads that earthquakes, hurricanes, tsunamis, and tornadoes frequently impose on our structures (Gardoni and LaFave, 2016). The experiments of today enable the infrastructure of tomorrow. As these transformational concepts are developed for infrastructure systems, the research community demands new testing platforms for dynamic experimentation that are realistic and cost-effective. Experimentation with and validation of such systems within the complex scenarios in which they will operate require a new generation of experimental platforms that are flexible, adaptive, predictive, and safe. As the range of scenarios involving dynamic experimentation

necessarily becomes more complex, it is clear that the experimental platforms required to conduct sufficiently deep investigations of these engineering systems do not yet exist (Condori et al., 2020).

Real-time hybrid simulation (RTHS, hereafter) is a powerful cyber-physical technique for dynamic experimentation that allows researchers to study the complex dynamical behaviors of infrastructure systems in realistic scenarios (Karavasilis et al., 2011; Gao et al., 2013). RTHS couples physical specimens and computational models in a single experiment to simulate a complete structural system. Transfer systems (e.g., hydraulic actuators and shake tables) are used to enforce interface conditions, numerical models are used to simulate the virtual components, and measurements are collected to feed into the numerical model. Control schemes are needed in this closed-loop system to compensate for transfer system dynamics and interactions with physical specimens, ensuring that the entire simulation is conducted safely and in the most realistic manner possible. The entire simulation is executed in real-time (i.e., the test duration is the duration of the extreme event) requiring strict timing guarantees.

Due to deep uncertainties in the physical substructure and transfer system, and the need for aggressive control to maintain stability and safety, advanced non-linear control, uncertainty quantification, estimation and prediction must be employed. Several of these issues have been examined in isolation and thus, RTHS has been a subject for continuous development over the past two decades. For instance, researchers have improved the accuracy and stability of interface condition enforcement, see (Gao et al., 2013; Ou et al., 2015; Palacio-Betancur and Gutierrez Soto, 2019; Tao and Mercan, 2019; Wang et al., 2019; Xu D. et al., 2019). Further, Chae et al. (2013), Chen et al. (2015), Maghareh et al. (2020), and Condori et al. (2020) developed adaptive actuator compensation schemes to achieve improved control of servo-hydraulic systems with non-linearities. Fermando and Spencer (2017) developed a multi-input, multi-output control design approach with accurate reference tracking and noise rejection. Wallace et al. (2005), Mercan and Ricles (2008), Maghareh et al. (2014), Maghareh et al. (2017), Gao and You (2019), and Xu W. et al. (2019) have also established and validated stability and accuracy metrics to enhance credibility and to encourage much broader applications of RTHS than have been possible to date, for example real-time aerodynamics hybrid simulation by Wu and Song (2019) and Su and Song (2019), experimental testing of spacecraft parachute deployment using RTHS by Harris and Christenson (2019). Nonetheless, despite the potential for using RTHS to conduct low-cost, high-efficiency experiments, the lack of a modular framework that can systematically integrate these existing capabilities, and do so while establishing clear requirements for the safety, integrity, and coordinated evolution of an RTHS experiment, has prevented researchers from tackling a broader range of complex problems.

This study presents the first phase of development and numerical validation of *Reflective Framework for Performance Management* (hereinafter referred to as REFORM) of real-time hybrid simulation. The main objective of this phase of development (REFORM-I) is to define the building blocks and

develop a modular architecture that will enable conducting black-box and reference-free experiments safely and with high confidence. A key aspect of this is having the ability to appropriately allocate dedicated resources to control and prediction tasks. In future phases of REFORM development, we will develop and share a highly modular framework capable of (1) exploiting existing prediction, control and model-updating techniques developed by different researchers in the RTHS research community, (2) adapting computational and control workloads and simulation rates, and (3) enabling more challenging and realistic experiments safely and with high confidence. We will make this available through an NSF-funded Research Coordination Network on this topic, known as the Multi-hazard Engineering Collaboratory on Hybrid Simulation (MECHS, <https://mechs.designsafe-ci.org/>).

FRAMEWORK DESCRIPTION

RTHS is intended to minimize the need for full-scale dynamic testing (e.g., shake table testing), and as such, these experiments must be performed without a full physical reference experiment. Thus, simulations are characterized by deep uncertainties in the physical substructure, significant coupling/interaction between the physical and computational substructures, and deep uncertainty in the emergent system behavior. When knowledge of the physical specimen, and therefore the reference structure, is limited, such *black-box* and *reference-free* testing necessitates the use of the latest developments in adaptive and robust control systems, prediction and estimation. REFORM-I is, therefore, broken down into five generic building blocks: (1) multi-rate coordination; (2) transfer system control; (3) state estimation and model updating; (4) run-time indicators; and (5) decision making. **Figure 1** shows an overview of REFORM-I representation and coordination between these building blocks.

Multi-Rate Coordination

In REFORM-I, a multi-rate coordination technique is required to enable users to coordinate the necessary multi-rate functionalities of the various building blocks. This multi-rate coordination block facilitates the use of complex and high-fidelity features. The Adaptive Multi-rate Interface (Maghareh et al., 2016) is utilized to meet this objective. The adaptive multi-rate interface (AMRI, hereafter) was initially developed as a mechanism to facilitate greater fidelity in the computational substructure by running the numerical model at a larger time-interval than what is used for the control system, thereby providing supporting use of a computationally demanding model. In REFORM-I, however, AMRI serves more generally as the multi-rate coordination technique that enable effective communications between feature blocks with different sampling rates, see **Figure 2**. In this method, after selecting a set of orthonormal bases (e.g., polynomial or exponential) sampling frequency ratio between the feature blocks. Then, a synchronized signal is generated by AMRI at the rate of Δt where X is the input signal at coarse time interval ΔT , Y is the output signal at sub-interval Δt , SFR is the sampling frequency ratio ($\Delta T/\Delta t$), k is the number of orthogonal bases used for interpolation, r is the number of points used for

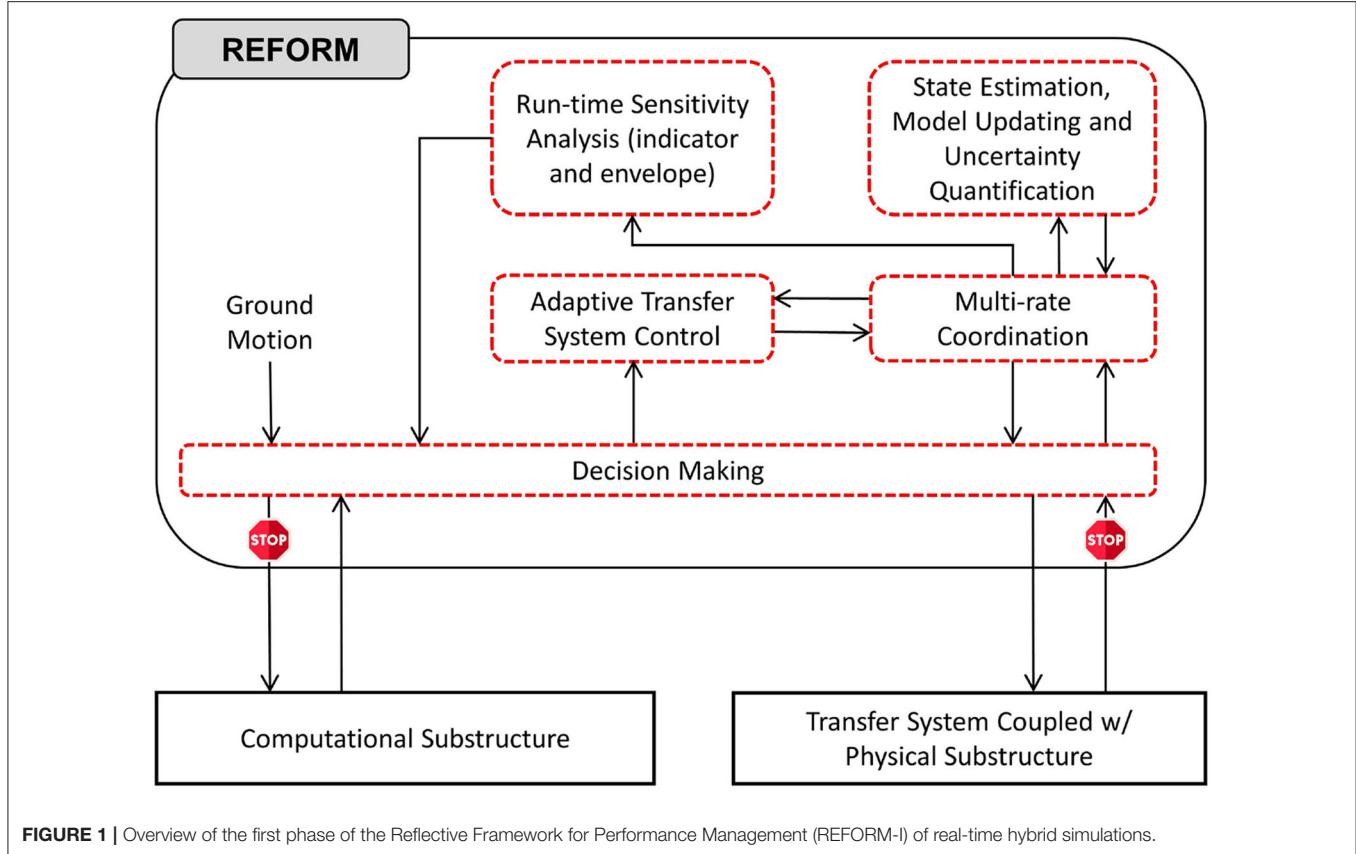


FIGURE 1 | Overview of the first phase of the Reflective Framework for Performance Management (REFORM-I) of real-time hybrid simulations.

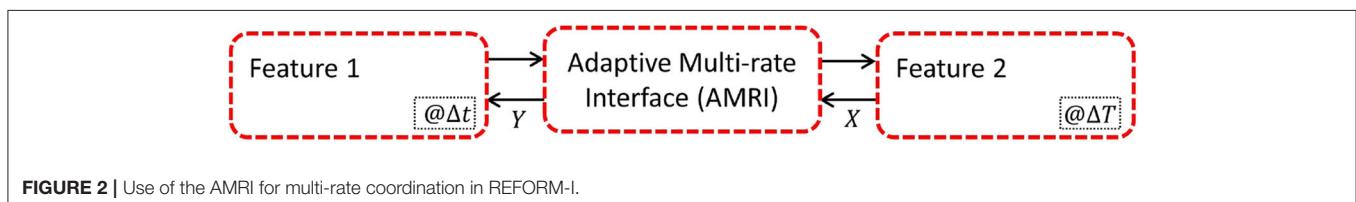


FIGURE 2 | Use of the AMRI for multi-rate coordination in REFORM-I.

interpolation, p is the compensation coefficient, and $p\Delta T$ is the time to be compensated.

Chebyshev polynomials of the first kind are used as the set of orthonormal bases for interpolation and rate transitioning from ΔT to Δt . The polynomials are defined by the following recurrence relation.

$$\begin{cases} T_1(s) = 1 \\ T_2(s) = s \\ T_{i+1}(s) = 2sT_i(s) - T_{i-1}(s) \end{cases} \quad (1)$$

These polynomials are adjusted to be within a general range of $[a, b]$, where, $a = (p + r - 1)\Delta T$ and $b = p\Delta T$. For this adjustment, $s = \frac{2x-(a+b)}{b-a}$ 100 where x corresponds to a dummy variable in the range of $[-1, 1]$, see **Figure 3**. Next, the following linear equation

is solved to obtain $\{\beta_1, \beta_2 \dots \beta_K\}$.

$$\begin{pmatrix} (T_1[(p+1-r)\Delta T] & \dots & T_k[(p+1-r)\Delta T]) \\ \vdots & \ddots & \vdots \\ T_1[(p)\Delta T] & \dots & T_k[(p)\Delta T] \end{pmatrix}_{r \times k} \times \begin{pmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_k \end{pmatrix} = \begin{pmatrix} X_{n-r+p+1} \\ \dots \\ X_n \\ \dots \\ X_{n+p} \end{pmatrix} \quad (2)$$

Using the β coefficients, the output signal at the coarse time interval Δt can be reconstructed as follows:

$$Y(h) = \beta_1 T_1(h) + \beta_2 T_2(h) + \dots + \beta_k T_k(h) \quad (3)$$

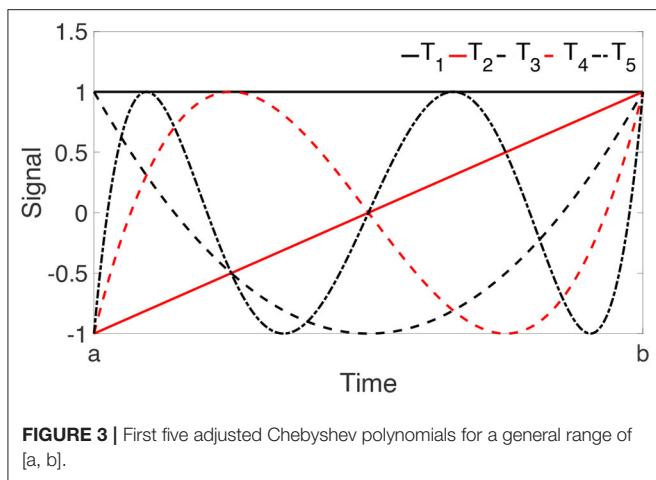


FIGURE 3 | First five adjusted Chebyshev polynomials for a general range of $[a, b]$.

where

$$h \in \{(n + p - 1)\Delta T, (n + p - 1)\Delta T + \Delta t, (n + p - 1)\Delta T + 2\Delta t \dots (n + p)\Delta T\}.$$

Adaptive Transfer System Control

This building block is broken down into two main tasks: (1) developing a physics-based, control-oriented non-linear dynamical model of a multi-actuator transfer system coupled with a non-linear physical specimen; and (2) designing a high-precision self-tuning robust controller to accommodate extensive variations in a non-linear control plant, such as non-stationary behavior or component failure.

In RTHS, controllability is a significant property of the transfer system. The key to modeling the control plant is making realistic assumptions within the operating range of an experiment, maintaining the essential dynamics, and discarding the insignificant ones. A fundamental step before developing a control strategy and designing a control law is realizing existing constraints and making realistic assumptions, for instance, the dynamic interactions between physical substructure and transfer system (a.k.a., control-structure interaction, Dyke et al., 1995) and extensive performance variations and uncertainties associated with the control plant (Maghreh et al., 2018a,b; Condori et al., 2020).

In REFORM-I, we have developed deterministic and stochastic physics-based non-linear dynamical models of a servo-hydraulic transfer system coupled with a non-linear physical specimen. These models have been developed for single- and multi-actuator systems and experimentally validated for a single actuator system coupled with a non-linear physical specimen (Maghreh et al., 2018a,b). Transforming the plant model into a controllable canonical dynamical model makes it appealing for developing more advanced non-linear control systems. Adopting these models becomes especially important in two cases: (1) when control-structure interaction dominates the dynamics of the coupled systems; and (2) when the

hydraulic system is coupled with a physical specimen with high uncertainty.

The physics-based, non-linear dynamical models of a servo-hydraulic transfer system coupled with a non-linear physical specimen serve as the basis for developing an effective control strategy which accommodate highly uncertain dynamics of the control plant and *control-structure interaction* (Maghreh et al., 2018a,b; Montoya et al., under review). To illustrate the capabilities required in this building block, we use a model-based, multi-layer nonlinear control system (*Self-tuning Robust Multi-directional Control System*, hereinafter referred to as SCRSys) developed by Maghreh et al. (2020) as the adaptive transfer system control strategy, see **Figure 4**. This control strategy has been developed, validated and incorporated in REFORM-I to accommodate extensive performance variations and uncertainties in the physical substructure. Our initial experiments with the Self-tuning Robust Multi-directional Control System also suggest that it has significant promise for complex multi-directional scenarios. SCRSys consists of two layers, robustness and adaptation. The robustness layer synthesizes a non-linear control law such that the closed-loop dynamics perform as intended under a broad range of parametric and non-parametric uncertainties. Then the adaptation layer reduces those uncertainties at run-time through slow and controlled learning of the control plant. SCRSys is developed for single- and multi-actuator transfer systems and experimentally validated for single-actuator RTHS experiments (Maghreh et al., 2020).

Here, the layer of adaption is designed to reduce estimation error, while the stability of the layer of robustness remains intact. In an RTHS experiment, there are parametric and non-parametric uncertainties associated with the control plant. Under certain conditions, the layer of robustness filters out the high frequency unmodeled dynamics in the transfer system (Maghreh et al., 2020). However, unless the parametric uncertainties are gradually reduced at run-time by a self-tuning mechanism, they may cause tracking performance degradation. Thus, the layer of adaptation is to suppress the parametric uncertainties, or the parametric variations—e.g., yielding or internal resonance in the physical substructure—or both, while tracking performance is consistently improving. The control and adaptation laws associated with SCRSys are provided in (Slotine and Li, 1991; Maghreh et al., 2020).

State Estimation, Model Updating, and Uncertainty Quantification

To facilitate run-time model updating and uncertainty quantification, and to eventually develop run-time performance indicators, strategies to improve computational efficiency are essential. RTHS experiments are frequently performed to study unexpected behavior or a part of the physical substructure may experience a failure, drastically changing its dynamics. Such a change will impact the safety and integrity of the experiment. Therefore, the run-time model updating and uncertainty quantification methods adopted must be able to handle such large variations in behavior. The remainder of this section

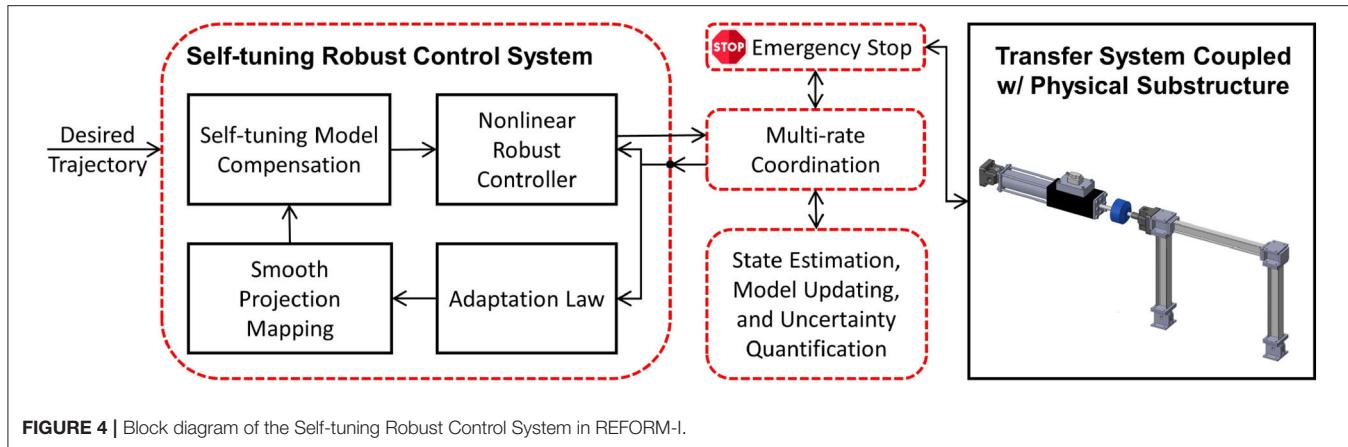


FIGURE 4 | Block diagram of the Self-tuning Robust Control System in REFORM-I.

presents the choices of the particle filter for model updating and uncertainty quantification in REFORM-I.

Various strategies are available for run-time model updating and uncertainty quantification. One conventional approach is to apply an optimization strategy to identify the parameters that minimize the difference between the predicted and measured model properties (e.g., residual-based model updating by Li et al., 2018). However, most of these model updating methods are deterministic, and do not consider uncertainties in the measurements or in the model adopted. In contrast, methods grounded in Bayesian inference determine the distribution of each of the uncertain model parameters based on observations (Zarate et al., 2012). Thus, they not only offer the ability to estimate the model parameters, but they directly facilitate rigorous uncertainty quantification for those parameters and/or states. However, due to high computational cost of such methods, and the lack of efficient sampling schemes using Monte Carlo simulation, they are quite limited for use during run-time in most RTHS experiments (Yang and Lam, 2018). Alternatively, various Bayesian filtering techniques have the potential to support both model updating and uncertainty quantification at run-time (Sarkka, 2013). Typical examples of such techniques include the extended Kalman filter, unscented Kalman filter, and particle filter. After initialization, this class of algorithms support predictive capabilities to estimate the distribution of model parameters or responses at each time step, and update the posterior distribution based on observed data.

The particle filter is a non-parametric, recursive Bayesian state estimator. It represents the posterior probability density function of the states as samples, or particles, with associated weights (Aru, 2007). In this building block, the particle filter is used for parameter estimation, uncertainty quantification, and response prediction, facilitating run-time model updating. The workflow of the particle filter block is illustrated in Figure 5. The particle filter is first initialized by assigning the number of particles and initial conditions (state and covariance). The particles are then sampled from a Gaussian distribution representing the prior density function. Afterwards, particles predict the next state based on an appropriate state transition function. For an illustrative example of the use of the particle filter for parameter

identification in highly non-linear systems (see Condori-Uribe et al., 2019).

To enable black-box RTHS and accommodate extensive variations in the physical specimen, such as non-stationary behavior or component failure, non-linear estimators (e.g., unscented Kalman filters, particle filters) are needed. Run-time execution of such filters enables: (1) estimation of the states and updating of the parameters of the physical specimen model for SRCSys implementation; and (2) monitoring, in real time, of the overall performance of the dynamic simulation. In REFORM-I, real-time non-linear estimators are developed by adopting a particle filter algorithm that takes advantage of measured displacement and force signals. The particle filter algorithm developed for this building block can serve multiple purposes including: parameter estimation (a.k.a. model updating), uncertainty quantification, and state estimation.

This approach is shown to significantly improve the performance of the real-time control system and accommodates extensive variations in the behavior of the physical specimen (Condori-Uribe et al., 2019). In addition, it enables run-time estimation and associated uncertainties of the model parameters, thus serving as the basis for *run-time sensitivity indicators* and *run-time sensitivity envelope*. However, including computationally-intensive estimation techniques does necessitate the use of the multi-rate coordination block for run-time execution. Further details on the state estimation building block for non-linear systems using the particle filter are provided in Condori-Uribe et al. (2019).

Run-Time Sensitivity Indicator and Stability Threshold

An essential building block in REFORM-I is the run-time thresholds and indicators to facilitate safe and accurate experiments. Stability and accuracy of a particular RTHS configuration are mainly functions of four factor: (1) overall dynamics of the reference structure; (2) fidelity of the computational substructure (related to computational workload and simulation rate); (3) partitioning configuration; and (4) capability to implement interface conditions (related to control workload and simulation rate) (Maghreh et al., 2017).

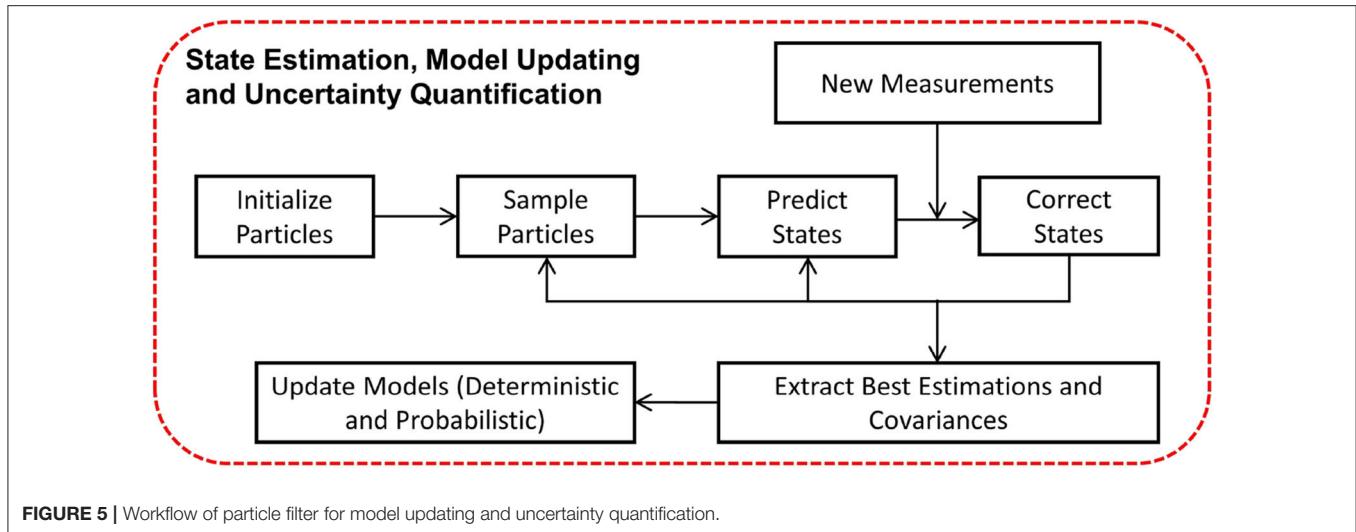


FIGURE 5 | Workflow of particle filter for model updating and uncertainty quantification.

In this building block, we have established run-time sensitivity indicator, run-time sensitivity envelope, and run-time stability threshold based on the first, third, and fourth factors. To compute the sensitivity indicator, envelope, and the stability threshold on-the-fly, we (1) use updated parameters and associated uncertainties of the physical substructure (from the non-linear estimator in the preceding section); (2) evaluate tracking performance at the interface; and (3) compute critical time delay.

In REFORM-I, we developed the run-time sensitivity indicator based on the knowledge gained in developing predictive (before any testing) stability and performance indicators intended to support the design of challenging RTHS experiments (Maghareh et al., 2014, 2017). For these predictive indicators, the sensitivity in partitioning of any linear or piecewise linear partitioned system to interface desynchronization is computed before the testing. The run-time sensitivity indicator (and run-time sensitivity envelope), however, are computed based on the sensitivity of the potential partitioning associated with the computational substructure and the estimated model (and associated uncertainties) of the physical substructure in real-time. To compute these in real-time, we: (1) characterize the potential critical time delay (i.e., the delay associated with the first occurrence of stability switch in **Figure 6**) associated with a wide and specific range for parametric variations ($\alpha_1, \alpha_2, \dots, \alpha_n$, see **Figure 7**) in the potential physical substructure model using the method developed in Maghareh et al. (2017); (2) update the physical substructure model in real-time using a particle filter; (3) update the uncertainties associated with these parameters; (4) compute, on the fly, the critical time delays (τ_{cr}) associated with virtual frameworks shown in **Figure 6**; and, (5) compute the run-time sensitivity indicator (RSI) and run-time envelope as follows:

$$RSI = -\log_{10}[\tau_{cr}(\text{msec})] \quad (4)$$

Equation (4) maps $\tau_{cr} \in (0, \infty)$ to the RSI $\in (-\infty, \infty)$. To calculate the run-time sensitivity envelope, however, a vector

of run-time sensitivity indicators (associated with the updated range of parametric variations in **Figure 6** obtained from the particle filter) are computed using Equation (4) and the envelope is constructed based on the maximum and minimum values at each time step.

The run-time stability threshold, however, evaluates tracking performance, in real-time, at the interface between the physical and computational substructures. This threshold is based on two factors: (1) the control strategy, and (2) the actual and instant sensitivity of the experiment. The latter is estimated using the RSI. The remainder of this section discusses the adopted strategy to monitor the run-time stability of the experiment using SCRSys.

In SCRSys, we have broken down the tasks associated with the non-linear tracking problem into two parts (Slotine and Li, 1991; Maghareh et al., 2020): (1) the control law is designed based on defining a compact tracking error. The compact tracking error creates a time-varying hyperplane called boundary layer. As a part of developing the control law, the time-varying boundary layer is designed to be an invariant set; (2) once the time-varying boundary layer is reached, the tracking error is determined, bounded and stability is guaranteed based on Barbalat's lemma (Slotine and Li, 1991).

The control law in the SCRSys is based on the assumption that the uncertainties associated with the dynamics of the plant are bounded. This assumption is aligned with our goal of determining the specific ranges of parametric variations in developing the RSI. In other words, the design of the SCRSys control law also specifies the ranges of parametric variations associated with the physical specimen. During an RTHS experiment, the boundary layer thickness and the compact tracking error over time convey a significant amount of information about whether the specific ranges of parametric variations are suitably chosen or not. A violation of this threshold, i.e., when the compact tracking error breaches the boundary layer, indicates that the stability of the control loop (and therefore the entire simulation) may be compromised. Thus, in REFORM-I the run-time stability threshold is a time-varying metric which is monitored in real-time and computed

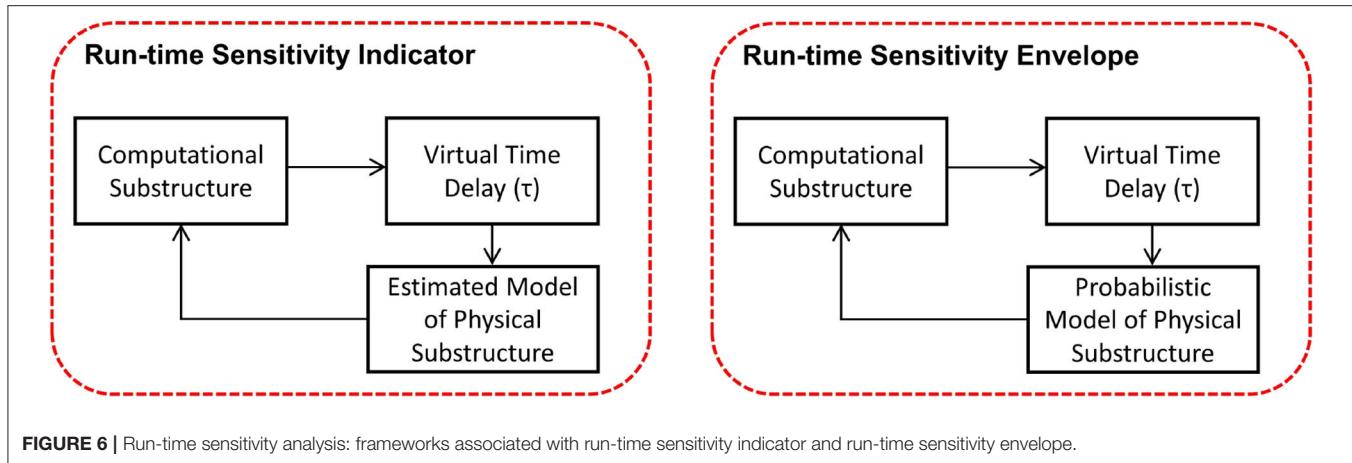


FIGURE 6 | Run-time sensitivity analysis: frameworks associated with run-time sensitivity indicator and run-time sensitivity envelope.

as the minimum distance between the compact tracking error and the boundary layer. **Figure 8** demonstrates the real-time computation and visualization of the run-time stability threshold.

REFORM-I is designed to provide a robust control framework without compromising the stability and safety of an RTHS experiment. Together, the run-time stability threshold, the run-time sensitivity indicator and envelope serve as the foundation for evaluating the integrity and safety of an RTHS experiment. During an RTHS experiment using REFORM-I, should the run-time stability threshold become negative, the emergency stop function would be activated and will decouple the computational and physical substructures through switching off the ground excitation signal and the feedback force from the physical substructure to the computational substructure. Thus, the entire simulation will become a simple tracking control problem which tracks a smooth decay signal which is associated with the unforced response of the computational substructure. The purpose of the emergency stop function is to ensure a stable experiment, suppressing any situation in which large actuators could potentially generate dangerous physical instabilities in the lab environment. During an experiment, run-time indicators are used to predict such behavior and trigger the emergency stop function. Upon the activation of the emergency stop, the computational substructure will be decoupled from the transfer system. At the same time, the command signal(s) to the hydraulic actuator(s) will follow a smooth decay signal(s) to a safe state.

NUMERICAL VALIDATION

The RTHS benchmark control problem (Silva et al., 2020) is employed for illustration and numerical validation of the building blocks that comprise REFORM-I. This section is structured as follows. First, we provide a brief overview of the benchmark problem. Next, we describe the three numerical studies considered in this section, and their corresponding virtual-RTHS (vRTHS, hereafter) results.

The Benchmark Control Problem for RTHS

The benchmark control problem considers a laboratory model of a typical frame structure. The model has three stories and two

bays, has pinned connections at the base, and moment-resisting connections with strong-column weak-beam design. A two-dimensional finite element (FE) model is developed using linear elastic models. Simplifying assumptions are made to reduce the order of the FE model from 30 DOF to 3 DOF, lowering the computational complexity of the model. The equation of motion for the reduced order model is referred to as the reference model,

$$M_r \ddot{x} + C_r \dot{x} + K_r x = -M_r t \ddot{x}_g \quad (5)$$

where M_r , C_r , and K_r are mass, stiffness, and damping matrices of the reference model, respectively. \ddot{x}_g denotes ground acceleration and x , \dot{x} and \ddot{x} are displacement, velocity, and acceleration vectors, respectively. The reference model is then partitioned into numerical and experimental substructures as shown in **Figure 9**. The red solid lines represent the experimental substructure. Accordingly, the partitioned equation of motion can be expressed as

$$M_n \ddot{x} + C_n \dot{x} + K_n x = -M_r t \ddot{x}_g - \underbrace{(M_e \ddot{x} + C_e \dot{x} + K_e x)}_{fe}. \quad (6)$$

where the $(\cdot)_n$ and $(\cdot)_e$ subscripts refer to the computational and physical substructures, respectively. System identification is performed using the experimental data to obtain the experimental structural mass ($m_e = 29.1$ kg), lateral stiffness ($k_e = 1.19 \times 10^6$ N/m), and damping coefficient ($c_e = 114.6$ N·s/m). Since the experimental substructure is fixed, several partitioning cases are defined by varying the structural parameters of the reference structure, as shown in **Table 1**. Variations are considered in both the modal damping and the mass of each floor of the reference structure, thus yielding different stability and performance scenarios. Using the predictive stability indicator and the proposed sensitivity classifications (Maghreh et al., 2017), the partitioning configurations associated with Cases 1–3 fall within the slightly sensitive class, while Case 4 falls within the moderately sensitive class. Thus, Cases 1–3 are less sensitive to desynchronization at the interface as compared to Case 4. Further, the predictive indicators reveal that when using the same transfer system control strategy, a researcher should expect somewhat less accuracy in the

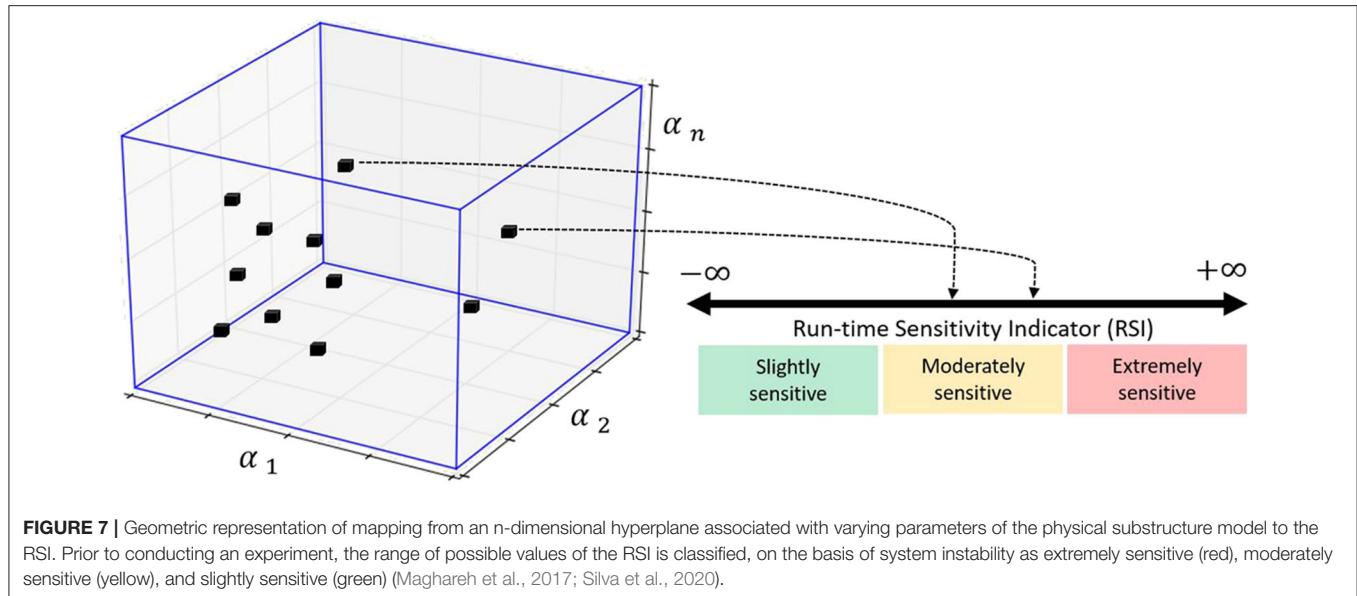


FIGURE 7 | Geometric representation of mapping from an n -dimensional hyperplane associated with varying parameters of the physical substructure model to the RSI. Prior to conducting an experiment, the range of possible values of the RSI is classified, on the basis of system instability as extremely sensitive (red), moderately sensitive (yellow), and slightly sensitive (green) (Maghareh et al., 2017; Silva et al., 2020).

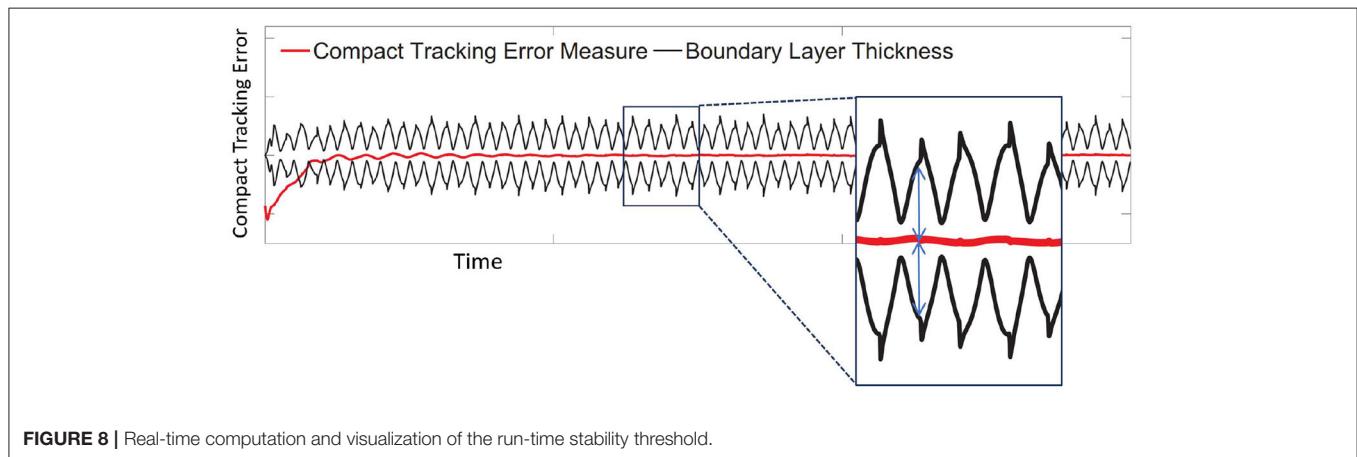


FIGURE 8 | Real-time computation and visualization of the run-time stability threshold.

results associated with Case 4 compared to the results from Cases 1–3. It should be noted that this paper only studies the extreme ones, Case 1 (i.e., the least sensitive case to desynchronization) and Case 4 (i.e., the most sensitive case to desynchronization).

Figure 10 shows the implementation of RTHS using a sample control strategy [i.e., a phase-lead compensator and a proportional-integral (PI) controller] provided by Silva et al. (2020) to demonstrate the problem. The ground input (\ddot{x}_g) and the feedback force vector (f_e) are the inputs of the numerical substructure whose 296 output (y_n) is fed into the tracking controller. The controller generates a command signal to the transfer system. The force at the first floor is measured and fed back to the numerical substructure to construct the force feedback loop (see Equation 6). The displacement output of the experimental substructure is used to form the tracking feedback loop.

Virtual RTHS Studies and Results

This section presents the numerical validation of the building blocks provided in REFORM-I. Because this phase is focused on developing a *modular framework* to enable conducting experiments *safely and with high confidence* by dedicating the appropriate resources to perform the control and prediction tasks, we perform three vRTHS studies using REFORM-I. Study 1 focuses on Cases 1 and 4 in the benchmark problem, as defined previously and in the original problem statement. Then in Studies 2 and 3 we consider a sudden component failure during a vRTHS of Cases 1 and 4. In Study 2 and Study 3, we artificially define a story drift threshold (4.4 mm) at which a failure of one of the experimental columns in the benchmark structure occurs (i.e., k_e drops by 50%). In Studies 1 and 2, we implement the sample linear control strategy provided in the benchmark problem. However, in Study 3, the building blocks provided in REFORM-I are activated to enhance the safety and integrity of the simulation. To quantitatively evaluate the performance of each study, we

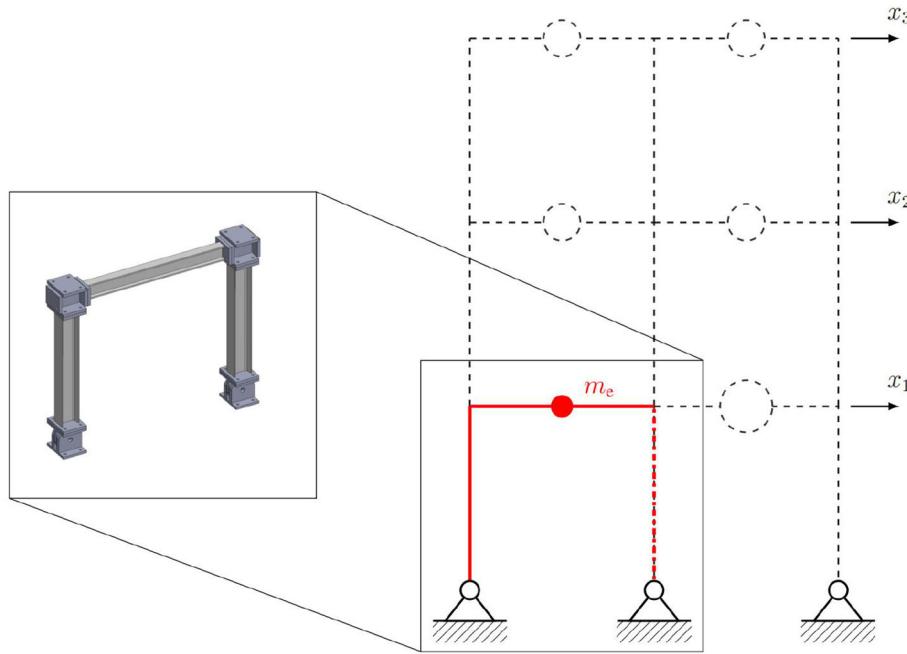


FIGURE 9 | Numerical and experimental substructures of benchmark structure by Silva et al. (2020).

utilize a cumulative normalized error indicator which is a global simulation performance indicator. The cumulative normalized error is defined as follows:

$$CNE_1(w) = \frac{1}{\max(|x_1^{\text{ref}}|)} \sum_{i=1}^w (|x_1^{\text{vRTHS}}(i) - x_1^{\text{ref}}(i)| \times \Delta t) \quad (7)$$

where CNE_1 , x_1^{vRTHS} , x_1^{ref} , and Δt refer to cumulative normalized error associated with the first floor, virtual-RTHS displacement at the first floor, reference displacement at the first floor, and sub-interval time step which is associated with displacement measurement. Note the cumulative normalized error is extremely sensitive to time desynchronization between the two signals which makes it a suitable error indicator for stability analysis of different systems and configurations.

Figures 11, 12 show the reference model responses, the vRTHS responses and cumulative normalized errors associated with Study 1 (Case 1) and Study 1 (Case 4), respectively. These results match those provided in the RTHS benchmark control problem by Silva et al. (2020). As noted earlier, the results show that the predictive stability indicator (PSI) associated with Case 1 ($PSI = 1.17$) is greater than that associated with Case 4 ($PSI = 0.94$), confirming that using the same transfer system control strategy leads to greater global accuracy in Case 1 as compared to Cases 4.

Figures 13, 14 show the reference model responses, the vRTHS responses and cumulative normalized errors associated with Study 2 (Case 1) and Study 2 (Case 4), respectively. In these simulations the lateral stiffness associated with the experimental substructure drops by 50% at 7.53 and 7.48 s,

TABLE 1 | Virtual-RTHS partitioning cases proposed by Silva et al. (2020).

Partitioning configuration	Reference floor mass (kg)	Reference modal damping (%)
Case 1	1,000	5
Case 2	1,100	4
Case 3	1,300	3
Case 4	1,000	3

respectively. The results confirm that Case 1 [PSI (before damage) = 1.17, PSI (after damage) = 1.50] is less sensitive to desynchronization at the interface as compared to Case 4 [PSI (before damage) = 0.94, PSI (after damage) = 1.25]. In addition, a comparison of the results between Study 1 and Study 2 shows that although the damage creates less sensitive configurations [compare PSI (before damage) and PSI (after damage)], the global performance of Study 1 is slightly more accurate than that of Study 2. This observation can be attributed to the fact that the sample linear controller provided in the RTHS benchmark problem definition lacks the level of robustness and adaptation required to accommodate extensive variations during a real-time hybrid simulation.

Figures 15–17 show the results of Study 3 (Case 1 and Case 4). In these simulations the lateral stiffness associated with the experimental substructures drops by 50% at 7.61 and 7.72 s, respectively. **Figure 15** shows the exact (off-line) sensitivity indicators, run-time sensitivity indicators, and run-time sensitivity envelopes associated with Cases 1 and 4. Prior

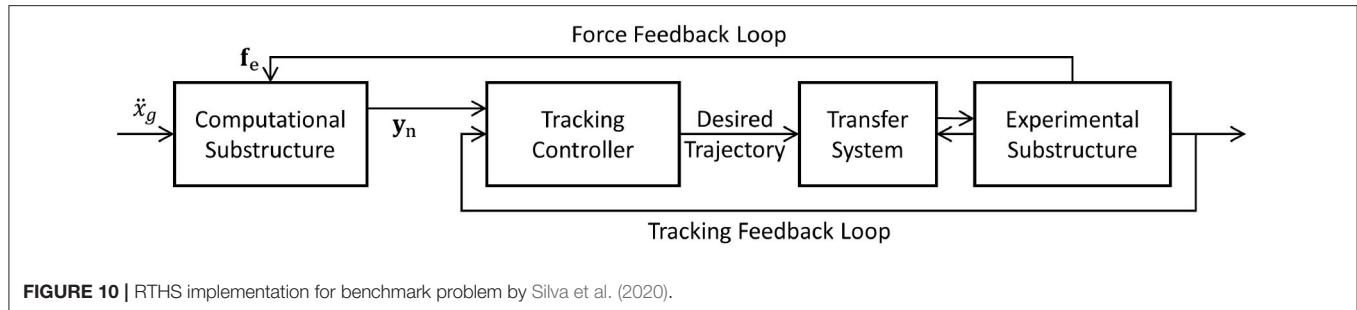


FIGURE 10 | RTHS implementation for benchmark problem by Silva et al. (2020).

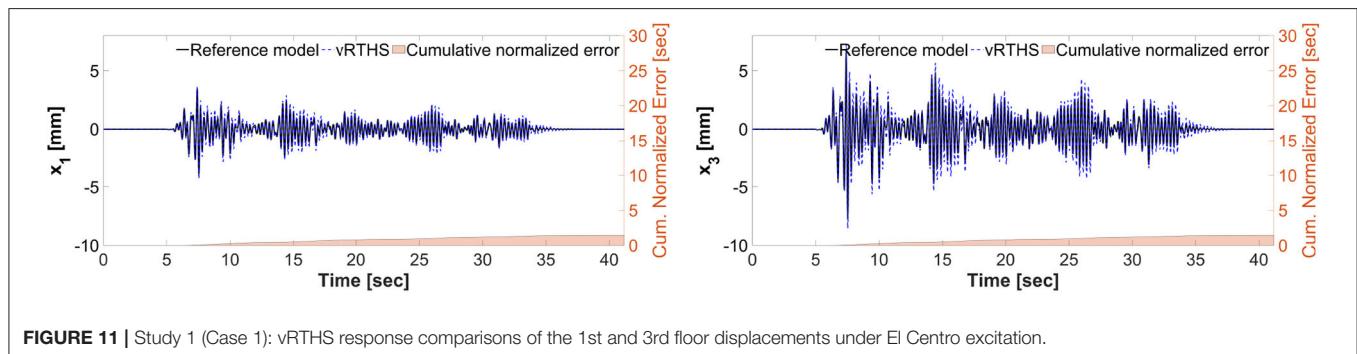


FIGURE 11 | Study 1 (Case 1): vRTHS response comparisons of the 1st and 3rd floor displacements under El Centro excitation.

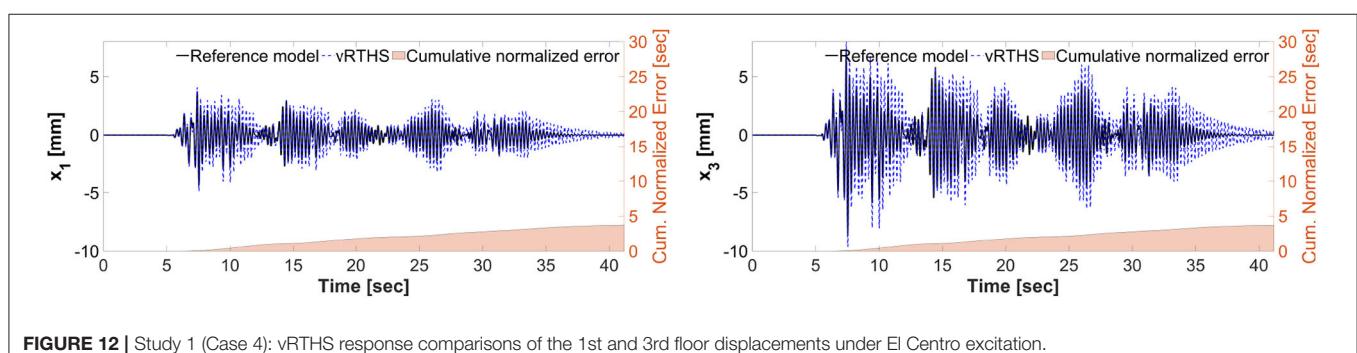


FIGURE 12 | Study 1 (Case 4): vRTHS response comparisons of the 1st and 3rd floor displacements under El Centro excitation.

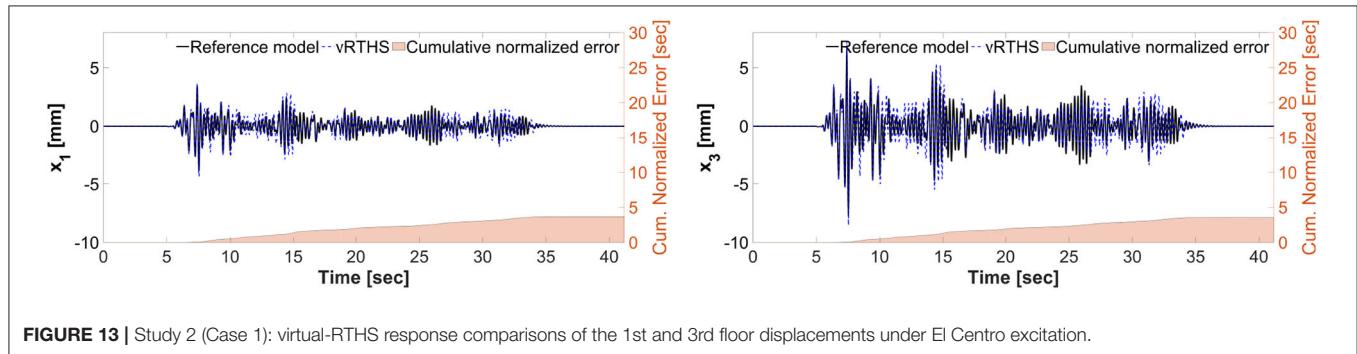


FIGURE 13 | Study 2 (Case 1): virtual-RTHS response comparisons of the 1st and 3rd floor displacements under El Centro excitation.

to conducting the simulations, the range of possible values of the RSI is classified, on the basis of system instability as extremely sensitive (red, $RSI > 0.7$), moderately sensitive (yellow, $0 \leq RSI \leq 0.7$), and slightly sensitive (green, $RSI < 0$), see Figures 6, 15.

These results also demonstrate that Case 1 is less sensitive to desynchronization at the interface both before and after the damage event. The exact sensitivity indicators are computed off-line using (1) the exact time of damage occurrence, and (2) full knowledge of the physical substructure parameters both

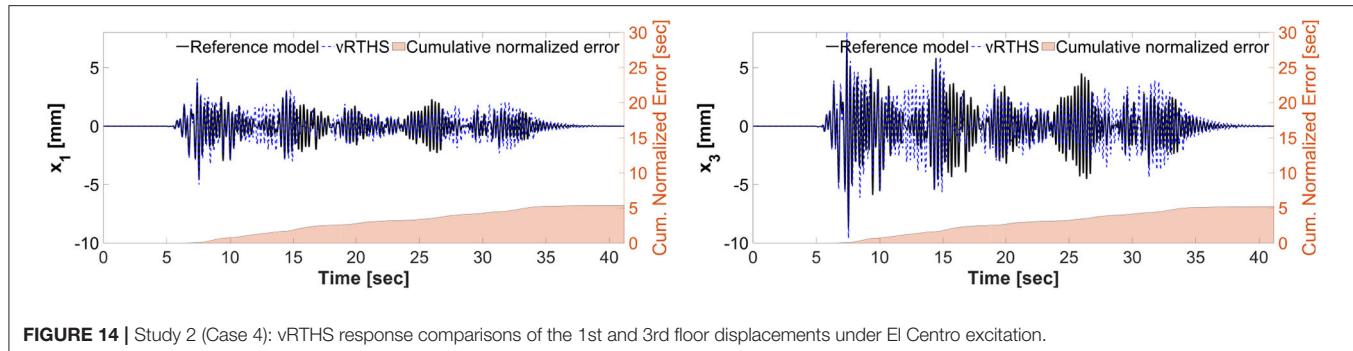


FIGURE 14 | Study 2 (Case 4): vRTHS response comparisons of the 1st and 3rd floor displacements under El Centro excitation.

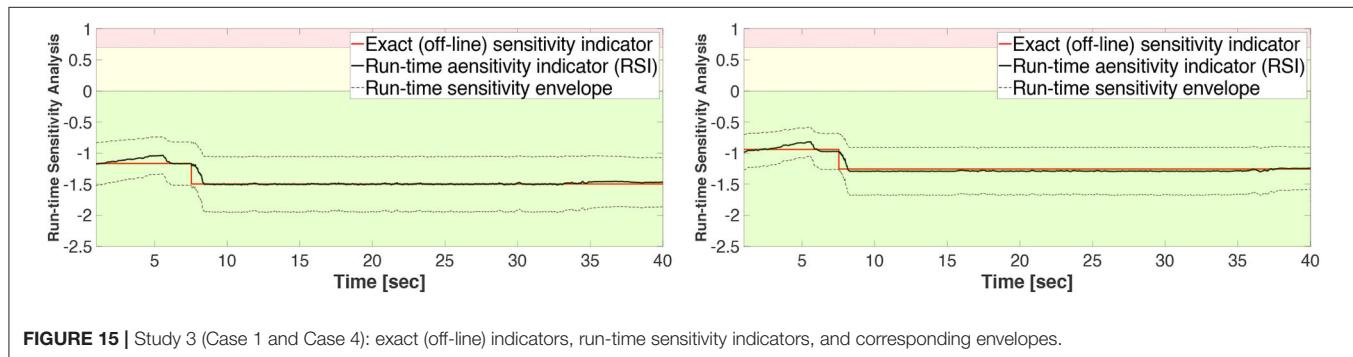


FIGURE 15 | Study 3 (Case 1 and Case 4): exact (off-line) indicators, run-time sensitivity indicators, and corresponding envelopes.

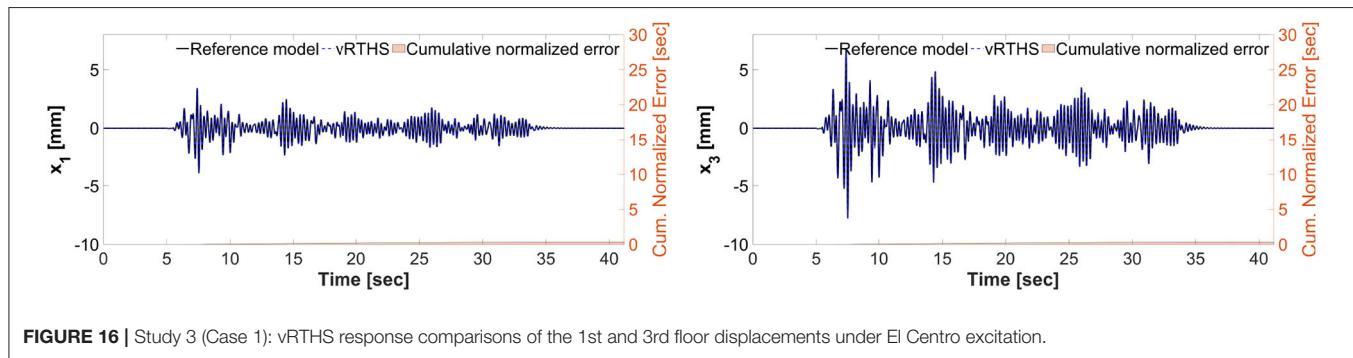


FIGURE 16 | Study 3 (Case 1): vRTHS response comparisons of the 1st and 3rd floor displacements under El Centro excitation.

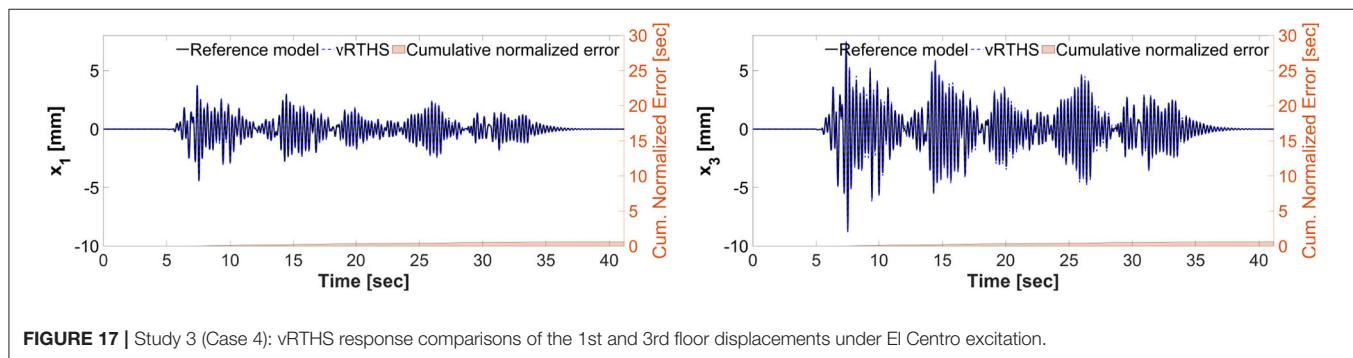


FIGURE 17 | Study 3 (Case 4): vRTHS response comparisons of the 1st and 3rd floor displacements under El Centro excitation.

before and after the damage occurrence. However, in an RTHS experiment neither will be available before the test begins. Therefore, we use (a) the model updating building block to estimate, on-the-fly, the parameters of the physical substructure

and (2) compute and monitor, in real-time, the sensitivities of the simulations before and after the occurrence of the damage. Aligned with our observation in Study 2, **Figure 17** also confirms that the failure of the column makes the simulation less sensitive

to desynchronization at the interface. However, unlike the poor global performance results associated with Study 2, the results of Study 3 provided in **Figures 16, 17** show significant improvements due to the built-in adaptation and robustness capabilities within REFORM-I.

Figure 15 shows that the exact sensitivity indicator falls within the run-time sensitivity envelope. This outcome is a critical requirement in case any extensive variations occur within the physical substructure (e.g., component failure, bifurcation, non-stationary dynamics, and dynamical switching), leading to more sensitivity to desynchronization at the interface. Whether or not the envelope can capture the exact sensitivity of the simulation will depend on the robustness of the run-time model updating and estimation building block.

CONCLUSIONS

Despite the potential for using RTHS to conduct cost-effective experiments at scale, the hazards engineering community has not been able to fully exploit this realistic testing method. The lack of systematic procedures and stringent requirements for the safety, integrity, and coordinated evolution of an RTHS test (during the experiment) and measures for the accuracy of experimental results (both during and after the experiment) have prevented researchers from tackling many problems that are of great interest to the hazards community. For instance, RTHS of systems with component damage or failure require advanced methods that can assess the system and adapt to changing parameters, necessitating run-time estimation of the new parameters and the associated uncertainty. In response to this technical hurdle, REFORM is developed to provide a modular framework that will enable conducting more challenging and realistic experiments safely and with high confidence. REFORM-I (this first phase of REFORM development and numerical validation) serves as a foundation for extending RTHS

application to black-box RTHS experiments: a general physical substructure with unanticipated dynamical behaviors. Herein we describe and demonstrate the building blocks and the numerical validation considered within REFORM-I. These building blocks (i.e., run-time sensitivity indicator, run-time stability threshold, run-time state-estimation and model updating technique, multi-rate coordination, and self-tuning robust control system) which are grounded in non-linear control and estimation theories, provide both mechanisms to support adaptation and robustness and the knowledge to use them effectively for conducting RTHS experiments. The methods are demonstrated and validated using a well-known benchmark problem in RTHS.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

AMa developed the self-tuning control system. YF and ZW developed the run-time model updating. JC developed the run-time state estimation. AMo and SD developed the validation problem. HM developed the run-time indicator.

FUNDING

This work was supported in part by a Space Technology Research Institutes grant (number 80NSSC19K1076) from NASA's Space Technology Research Grants Program, the Resilient Extraterrestrial Habitats project, funded by the Purdue University Office of the Provost (New Horizons program), the Research Coordination Network in Hybrid Simulation for Multi-hazard Engineering through NSF-CMMI 1661621, and the Purdue University Systems Collaboratory Fellows Program.

REFERENCES

Aru, Y. (2007). A tutorial on particle filters for online nonlinear/nongaussian bayesian tracking. *Bayesian Bounds Parameter Estimation Nonlinear Filter. Track.* 50, 723–737. doi: 10.1109/978047044198

Chae, Y., Kazemibidokhti, K., and Ricles, J. M. (2013). Adaptive time series compensator for delay compensation of servo-hydraulic actuator systems for real-time hybrid simulation. *Earthquake Eng. Struct. Dyn.* 42, 1697–1715. doi: 10.1002/eqe.2294

Chen, P.-C., Chang, C.-M., Spencer, B. F., and Tsai, K.-C. (2015). Adaptive model-based tracking control for real-time hybrid simulation. *Bull. Earthquake Eng.* 13, 1633–1653. doi: 10.1007/s10518-014-9681-2

Condori, J., Maghareh, A., Orr, J., Li, H.-W., Montoya, H., Dyke, S., et al. (2020). Exploiting parallel computing to control uncertain nonlinear systems in real-time. *Exp. Techn.* doi: 10.1007/s40799-020-00373-w

Condori-Uribe, J., Maghareh, A., and Dyke, S. J. (2019). Estimating states for nonlinear systems using the particle filter. *arXiv [Preprint]. arXiv:1911.03263*.

Dyke, S. J., Spencer, B. F., Quast, P., and Sain, M. K. (1995). Role of control-structure interaction in protective system design. *J. Eng. Mech.* 121, 322–338. doi: 10.1061/(ASCE)0733-9399(1995)121:2(322)

Fernandois, G. A., and Spencer, B. F. (2017). Model-based framework for multi-axial real-time hybrid simulation testing. *Earthquake Eng. Eng. Vibrat.* 16, 671–691. doi: 10.1007/s11803-017-0407-8

Gao, X., Castaneda, N., and Dyke, S. J. (2013). Real time hybrid simulation: from dynamic system, motion control to experimental error. *Earthquake Eng. Struct. Dyn.* 42, 815–832. doi: 10.1002/eqe.2246

Gao, X. S., and You, S. (2019). Dynamical stability analysis of MDOF real-time hybrid system. *Mech. Syst. Signal Process.* 133:106261. doi: 10.1016/j.ymssp.2019.106261

Gardoni, P., and LaFave, J. M. (2016). “Multi -hazard approaches to civil infrastructure engineering: mitigating risks and promoting resilience,” in *Multi-hazard Approaches to Civil Infrastructure Engineering*, eds P. Gardoni and J. M. LaFave (Cham: Springer International Publishing), 3–12. doi: 10.1007/978-3-319-29713-2

Harris, M. J., and Christenson, R. E. (2019). “Experimental test of spacecraft parachute deployment using real-time hybrid substructuring,” in *Sensors and Instrumentation, Aircraft/Aerospace and Energy Harvesting*, eds E. Wee Sit, C. Walber, P. Walter, A. Wicks, and S. Seidlitz (Cham: Springer), 67–70. doi: 10.1007/978-3-319-74642-5_8

Karavasilis, T. L., Ricles, J. M., Sause, R., and Chen, C. (2011). Experimental evaluation of the seismic performance of steel MRFs

with compressed elastomer dampers using large-scale real-time hybrid simulation. *Eng. Struct.* 33, 1859–1869. doi: 10.1016/j.engstruct.2011.01.032

Li, D., Dong, X., and Wang, Y. (2018). Model updating using sum of squares (SOS) optimization to minimize modal dynamic residuals. *Struct. Control Health Monitor.* 25, 1–16. doi: 10.1002/417 stc.2263

Maghareh, A., Dyke, S., Rabenjeharabbar, S., and Prakash, A. (2017). Predictive stability indicator: a novel approach to configuring a real-time hybrid simulation. *Earthquake Eng. Struct. Dyn.* 46, 95–116. doi: 10.1002/eqe.2775

Maghareh, A., Dyke, S. J., Prakash, A., and Bunting, G. B. (2014). Establishing a predictive performance indicator for real-time hybrid simulation. *Earthquake Eng. Struct. Dyn.* 43, 2299–2318. doi: 10.1002/eqe.2448

Maghareh, A., Dyke, S. J., and Silva, C. E. (2020). A self-tuning robust control system for nonlinear real-time hybrid simulation. *Earthquake Eng. Struct. Dyn.* 49, 695–715. doi: 10.1002/eqe.3260

Maghareh, A., Silva, C. E., and Dyke, S. J. (2018a). Parametric model of servo-hydraulic actuator coupled with a nonlinear system: experimental validation. *Mech. Syst. Signal Process.* 104, 663–672. doi: 10.1016/j.ymssp.2017.11.009

Maghareh, A., Silva, C. E., and Dyke, S. J. (2018b). Servo-hydraulic actuator in controllable canonical form: Identification and experimental validation. *Mech. Syst. Signal Process.* 100, 398–414. doi: 10.1016/j.ymssp.2017.07.022

Maghareh, A., Waldbjorn, J. P., Dyke, S. J., Prakash, A., and Ozdagli, A. I. (2016). Adaptive multi-rate interface: development and experimental verification for real-time hybrid simulation. *Earthquake Eng. Struct. Dyn.* 45, 1411–1425. doi: 10.1002/eqe.2713

Mercan, O., and Ricles, J. M. (2008). Stability analysis for real-time pseudodynamic and hybrid pseudodynamic testing with multiple sources of delay. *Earthquake Eng. Struct. Dyn.* 37, 1269–1293. doi: 10.1002/eqe.814

Ou, G., Ozdagli, A. I., Dyke, S. J., and Wu, B. (2015). Robust integrated actuator control: experimental verification and real-time hybrid-simulation implementation. *Earthquake Eng. Struct. Dyn.* 44, 441–460. doi: 10.1002/eqe.2479

Palacio-Betancur, A., and Gutierrez Soto, M. (2019). Adaptive tracking control for real-time hybrid simulation of structures subjected to seismic loading. *Mech. Syst. Signal Process.* 134, 106345. doi: 10.1016/j.ymssp.2019.106345

Sarkka, S. (2013). Bayesian filtering and smoothing. New York, NY: Cambridge University Press. Available online at: <https://books.google.com/books?hl=en&lr=&id=5VlsAAAAQBAJ&oi=fnd&pg=PR9&dq=sarkka+bayesian+filtering&ots=OPPSDASW7B&sig=IYItSwUiS7VE04fK8IGDq4qWgxU#v=onepage&q=sarkka%20bayesian%20filtering&f=false>

Silva, C. E., Gomez, D., Maghareh, A., Dyke, S. J., and Spencer, B. F. (2020). Benchmark control problem for real-time hybrid simulation. *Mech. Syst. Process.* 135:106381. doi: 10.1016/j.ymssp.2019.106381

Slotine, J. E., and Li, W. (1991). *Applied Nonlinear Control*. Englewood Cliffs, NJ: Pearson Education.

Su, W., and Song, W. (2019). A real-time hybrid aeroelastic simulation platform for flexible wings. *Aerospace Sci. Technol.* 95, 105513. doi: 10.1016/j.ast.2019.105513

Tao, J., and Mercan, O. (2019). A study on a benchmark control problem for real-time hybrid simulation with a tracking error-based adaptive compensator combined with a supplementary proportional-integral derivative controller. *Mech. Syst. Signal Process.* 134:106346. doi: 10.1016/j.ymssp.2019.106346

Wallace, M. I., Sieber, J., Neild, S. A., Wagg, D. J., and Krauskopf, B. (2005). Stability analysis of real-time dynamic substructuring using delay differential equation models. *Earthquake Eng. Struct. Dyn.* 34, 1817–1832. doi: 10.1002/eqe.513

Wang, Z., Ning, X., Xu, G., Zhou, H., and Wu, B. (2019). High performance compensation using an adaptive strategy for real-time hybrid simulation. *Mech. Syst. Signal Process.* 133:106262. doi: 10.1016/j.ymssp.2019.106262

Wu, T., and Song, W. (2019). Real-time aerodynamics hybrid simulation: wind-induced effects on a reduced-scale building equipped with full-scale dampers. *J. Wind Eng. Indus. Aerodyn.* 190, 1–9. doi: 10.1016/j.jweia.2019.04.005

Xu, D., Zhou, H., Shao, X., and Wang, T. (2019). Performance study of sliding mode controller with improved adaptive polynomial-based forward prediction. *Mech. Syst. Signal Process.* 133:106263. doi: 10.1016/j.ymssp.2019.106263

Xu, W., Chen, C., Guo, T., and Chen, M. (2019). Evaluation of frequency evaluation index based compensation for benchmark study in real-time hybrid simulation. *Mech. Syst. Signal Process.* 130, 649–663. doi: 10.1016/j.ymssp.2019.05.039

Yang, J. H., and Lam, H. F. (2018). An efficient adaptive sequential Monte Carlo method for Bayesian model updating and damage detection. *Struct. Control Health Monitor.* 25, 6–11. doi: 10.1002/stc.2260

Zarate, B. A., Caicedo, J. M., Yu, J., and Ziehl, P. (2012). Bayesian model updating and prognosis of fatigue crack growth. *Eng. Struct.* 45, 53–61. doi: 10.1016/j.engstruct.2012.06.012

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Maghareh, Fu, Montoya, Condori, Wang, Dyke and Montoya. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.